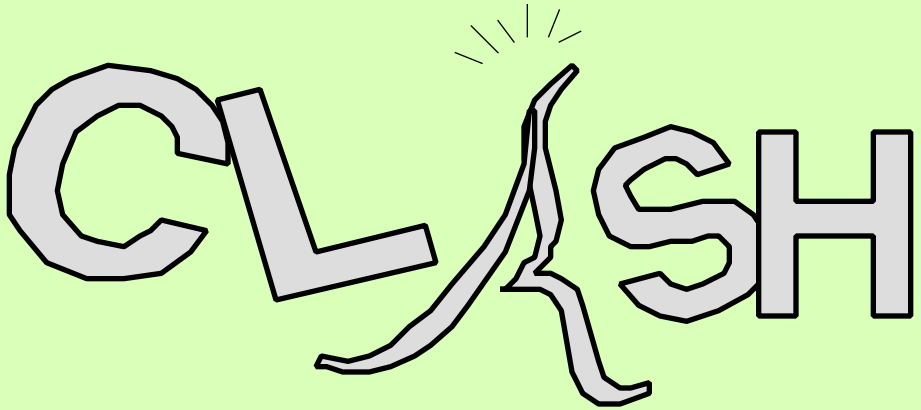
CRASH



tutorial

Bob Kanefsky

What is Clash?

- “C & Lisp Abstract Syntax Harmony”
- A way to compile both C and Lisp data structures from the same source files.
- A language for declaring message names
- A C code generator that hides boilerplate details
  - » freeing data allocated by another function
  - » “unmarshalling” bytes into structures
  - » type casting
- Written for DS1, implemented in Lisp.
- Called from command line or loaded into Lisp.

CLASH

Related topics to cover

- Marshalling
- Uses of Clash
  - » Compiling flight software's IPC code (C and Lisp)
  - » Compiling flight software's telemetry code (Lisp only)
  - » Converting scripts to ground commands
  - » Generating message-related test tools

CLASH

Definition of terms

IPC

Inter-Process Communications package

by Reid Simmons

used by C and Lisp FSW

marshal

to place bytes from a data structure in a standard order for transmission

unmarshal

to place bytes from an incoming stream into a data structure in a machine dependent, compiler-dependent order

Clash Preprocessor

Used for each C and Lisp module, to turn .clash files into .h files for C

Optionally used by C modules to generate wrappers to hide IPC-specific code

Clash Library for Lisp

A Lisp module, used mostly at compiletime for reading .clash and .h files.

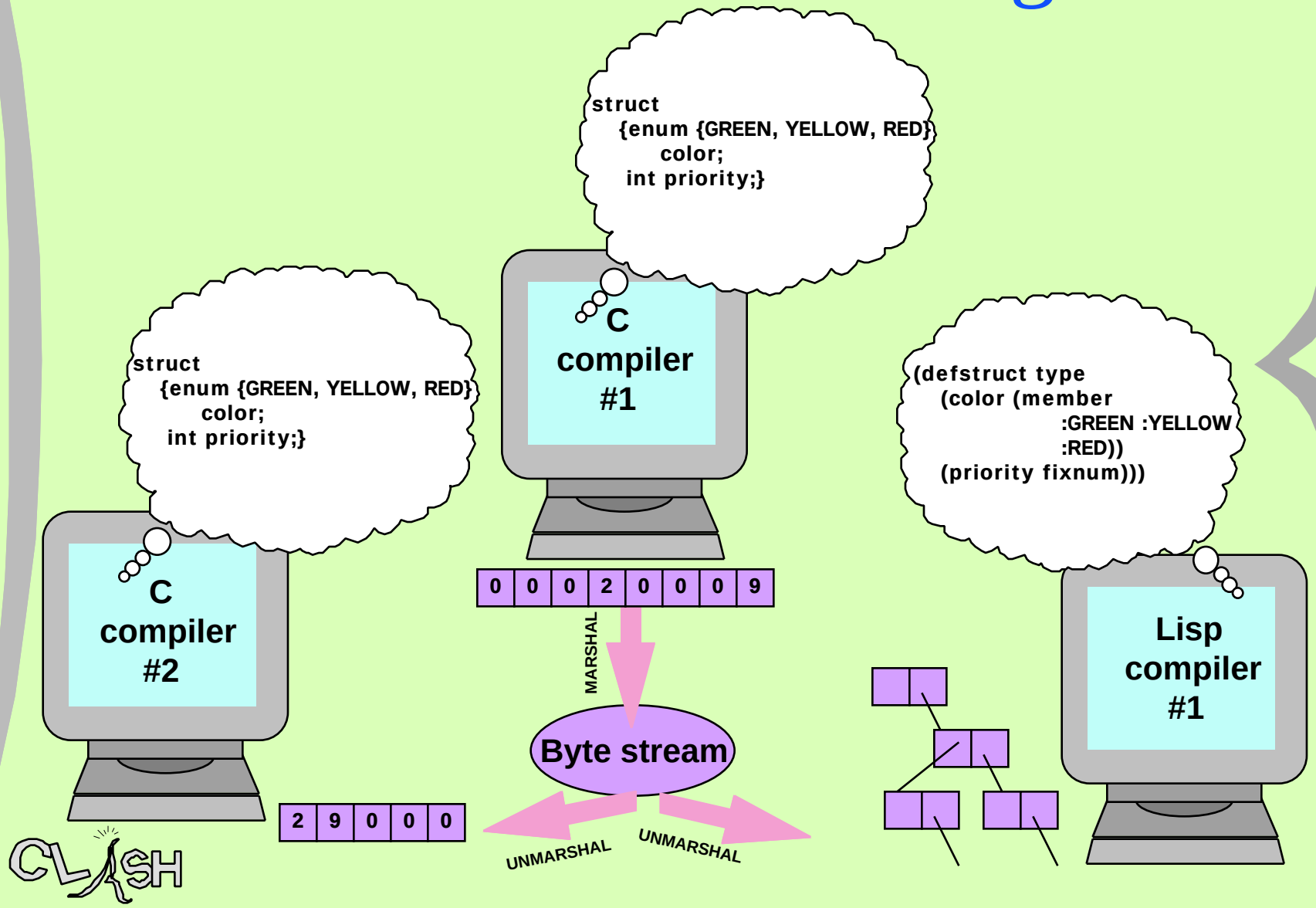
clomp

Any Clash-output Messaging Program (usually in C)

Independencies

- Marshalling can be used for more than IPC
- Clash can be used for more than marshalling and IPC

What is “marshalling”?



Clash takes care of the parallel definitions

.clash

source code written by user

```
(defargtype foo ()
   (structure
   (color (enumerated (GREEN yellow red)))
   (priority (integer :min 0 :max 1000000))))
```

generates:

```
typedef struct
   {enum {GREEN, YELLOW, RED}
      color;
    int priority;} foo
```

.h

```
(defstruct foo
   (color (member
               :GREEN :YELLOW
               :RED))
   (priority fixnum)))
```

Lisp

```
{{enum GREEN,YELLOW,RED},int}
```

format string

- Clash preprocessor converts .clash to .h
- Clash library for Lisp can read either .clash or .h directly at compiletime.

Sample use of Clash

```
(defargtype size_name (enumerated (LITTLE GIANT)))
(defargtype color_name (enumerated (RED YELLOW GREEN BLUE PURPLE))
(defargtype idenfication_type (enumerated  (ROCKS MEN WOMEN CVENTIPEDES)))
(defmessage PICTURE_CONTENTS_IDENTIFIED ((count (integer :maximum 1000))
                                         (size size_name) (color color_name)
                                         (type idenfication_type))
```

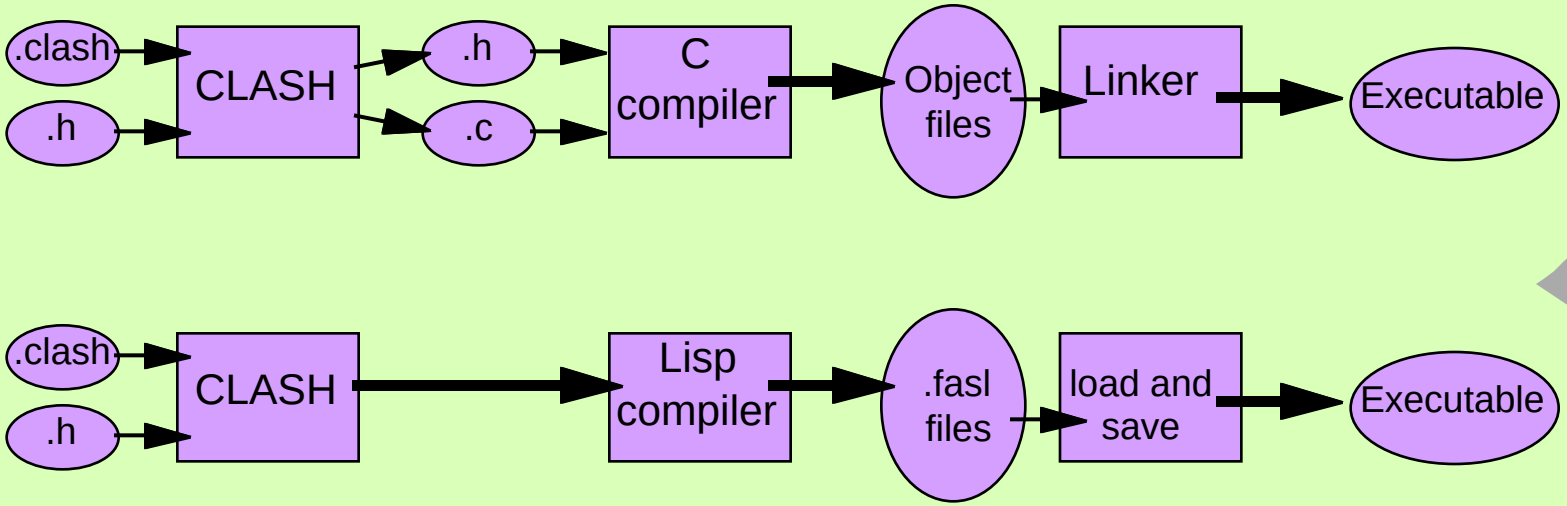
```
CLASH_declare_publishers (IMAGE_RECOGNIZER)
                PICTURE_CONTENTS_IDENTIFIED
                PICTURE_ERROR)
CLASH_declare_handlers (SMART_EXECUTIVE)
                PICTURE_CONTENTS_IDENTIFIED
```

Then write normal C or Lisp functions:

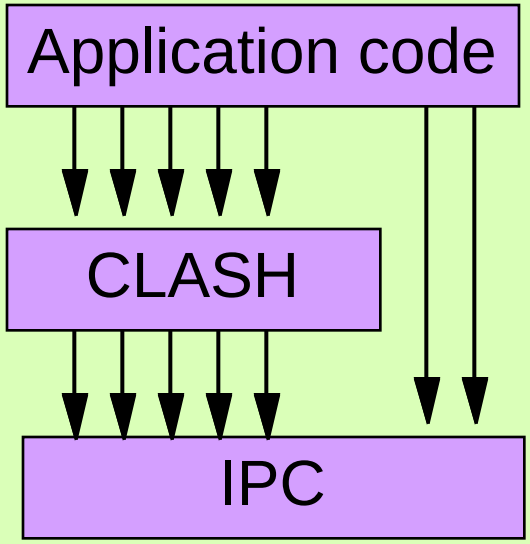
```
extern IMAGE_RECOGNIZER_send_PICTURE_CONTENTS_IDENTIFIED
                (short count, size_name size, color_name color,  idenfication_type type);
IMAGE_RECOGNIZER_send_PICTURE_CONTENTS_IDENTIFIED (3, LITTLE, GREEN, MEN);
```

```
(defun SMART_EXECUTIVE_handle_PICTURE_CONTENTS_IDENTIFIED (count size color type)
  (when (dangerous-monster-p size color type) (run-like-heck))
  (unless (equal type :ROCK) (set-beacon-mode *tone-for-interesting*))
```


Clash is a C preprocessor & Lisp macro package



Clash is a functional layer on top of IPC library calls



Clash for telemetry definitions

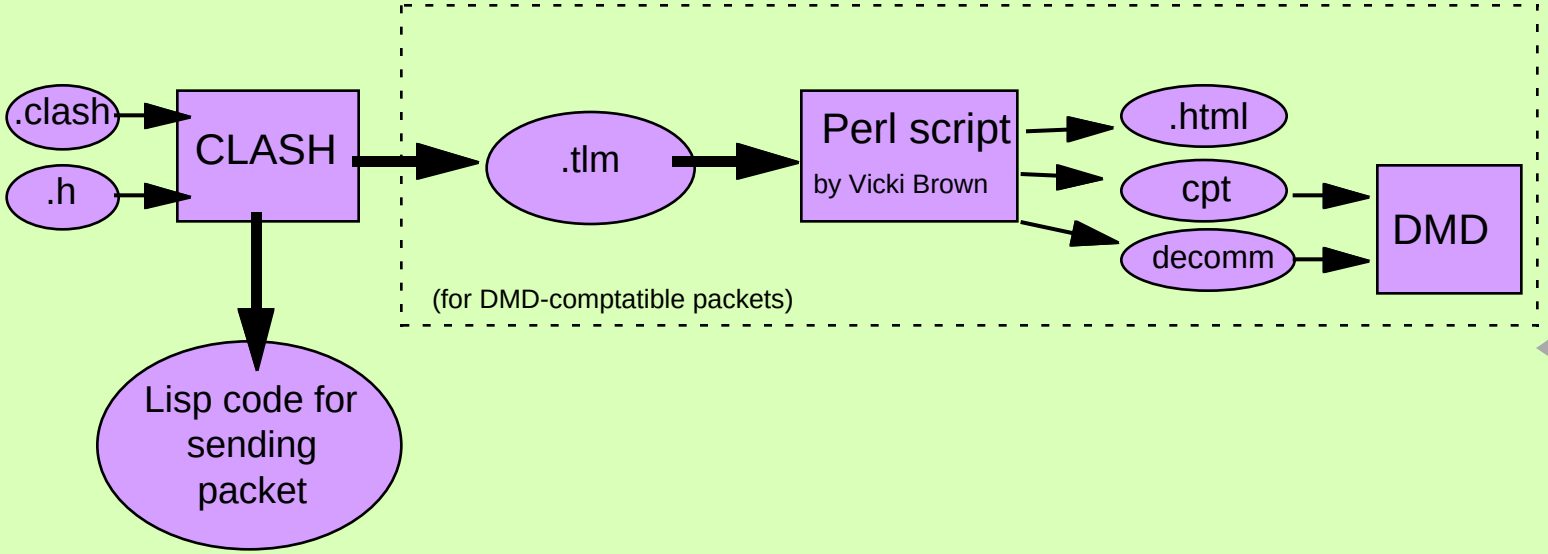
- Abstraction makes it easy to take a definition intended for use with IPC and re-use it for another type of communication.
  - » Here's the entire programming task for a Lisp programmer who wants to send a telemetry packet that corresponds to an existing IPC message:

defmessage-**with-telemetry**

defpublisher**-with-telemetry**

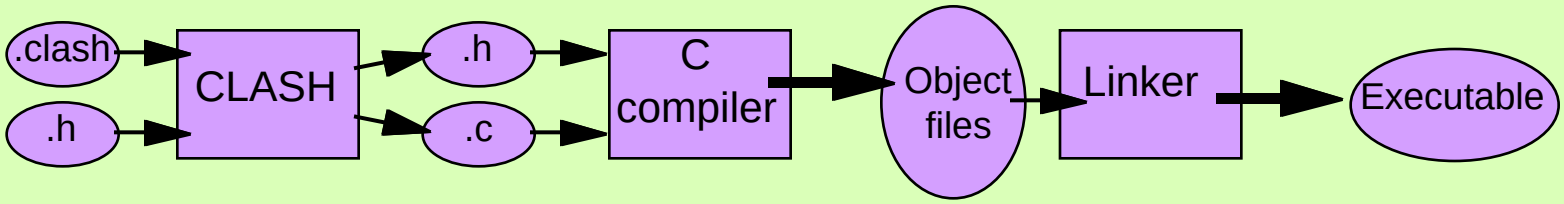
CLASH

Lisp telemetry interface



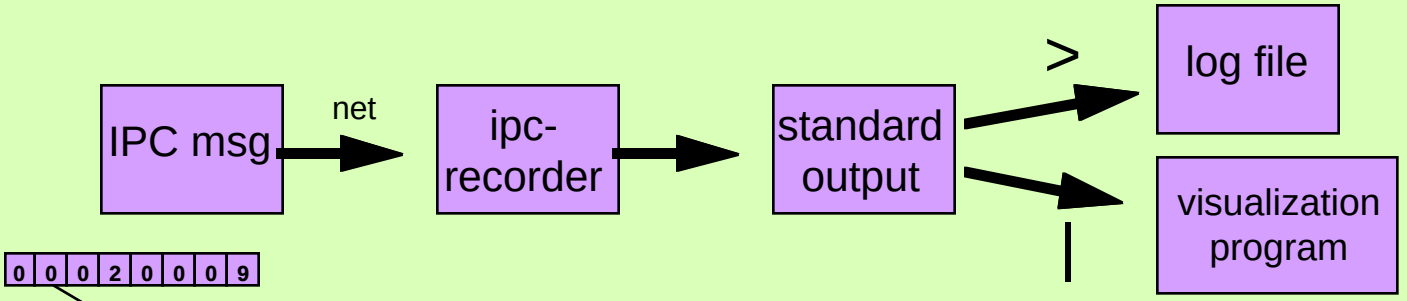
About clomps

- Given the message definitions, Clash can generate a clomp for printing all messages when received, replaying messages from a log file, or anything else that follows from the message definitions.



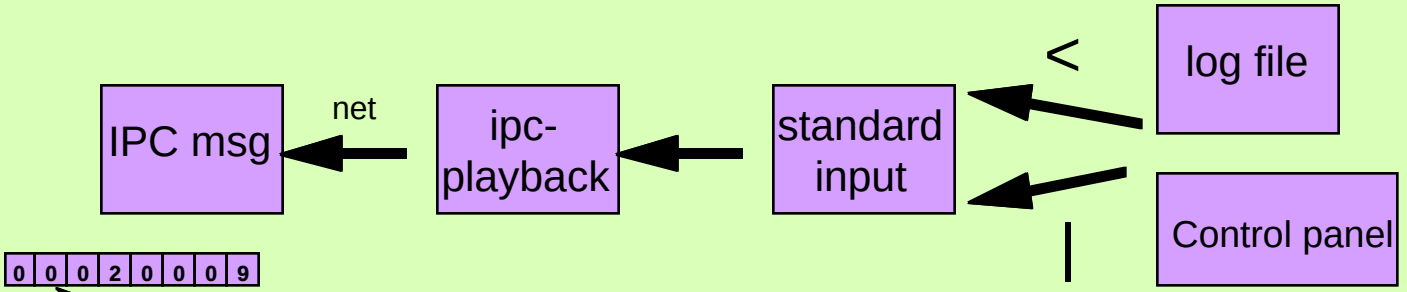
ipc-recorder clomp

- Receives IPC messages
- Outputs a printed representation to standard output
- Applications
  - » Save a log file
  - » IPC front-end for non-IPC-aware visualization programs



ipc-playback clomp

- Inputs a printed representation from standard input
- Sends IPC messages
- Applications
  - » Regression testing (also waits for selected messages)
  - » IPC front-end for non-IPC-aware control panels



write-script-to-binary-files clomp

- A proposed interface for sending ground commands
- Input: same format as ipc-recorder produces
- Create input by any method:
  - » sending ipc-messages
  - » GUI for composing messages (hand-written or future clomp)
  - » text editor
- Output: for each message, a binary file with marshalled message contents
- Data flow: just like ipc-playback, but writes files instead of sending IPC messages.

CLASH

Where to get more information

- DS1 Software **http://nmp.arc.nasa.gov/Entries/100/001/001/**
  - FSW Integration & Test **/Entries/906/001/319/**
    - CLASH Information **/Entries/994/660/382/**
    - Message-related Tools **/Entries/785/145/792/**